Evolution of pgAdmin: pgAdmin4

- Dave Page | 3 December 2016

About me

- PostgreSQL:
  - pgAdmin Project Lead
  - Core Team member
  - Web & Sysadmin Teams
  - Director (secretary) of PostgreSQL Europe
  - Chairman of PostgreSQL Community Association of Canada

- EDB:
  - Vice President & Chief Architect, Tools & Installers

pgAdmin III

- What is it?
- What was wrong with it?
- How do we fix it?

What is it?

- The leading Open Source GUI management tool for PostgreSQL
- Third generation (2002); replacing earlier tools written in VB (1998, 2001)
- Written in C++, using the wxWidgets cross-platform framework
- Ships standalone, and with the EDB ‘one-click’ PostgreSQL installers
- Supports PostgreSQL derivatives; EDB Postgres Advanced Server, and Greenplum Database

What is it?

pgAdmin III

Object browser

- Server Groups
  - Servers (6)
    - PG93 (local:/tmp/.s.PGSQL.5434)
    - PPAS 9.4 (127.0.0.1:5444)
    - PostgreSQL 9.2 (localhost:5432)
    - PostgreSQL 9.3 (localhost:5433)
      - Databases (5)
        - ascii
        - gleu
        - pem
        - pem4
          - Catalogs (2)
          - Casts (0)
          - Extensions (1)
          - Foreign Data Wrappers (0)
          - Languages (1)
          - Schemas (4)
            - pem
              - Aggregates (0)
              - Collations (0)
              - Conversions (0)
              - Domains (0)
              - Foreign Tables (0)
              - FTS Configurations (0)
              - FTS Dictionaries (0)
              - FTS Parsers (0)
              - FTS Templates (0)

Properties | Statistics | Dependencies | Dependents

Property	Value
Name	create_mib_notification_type
OID	27171
Owner	postgres
Argument count	2
Arguments	object_type integer, group_oid_type integer
Signature arguments	integer, integer
Return type	text
Language	plpgsql
Returns a set?	No
Source	DECLARE...
Estimated cost	100
Volatility	VOLATILE
Leak proof?	No

SQL pane

```
-- Function: pem.create_mib_notification_type(integer, integer)

-- DROP FUNCTION pem.create_mib_notification_type(integer, integer);

CREATE OR REPLACE FUNCTION pem.create_mib_notification_type(object_type integer, group_oid_type i
  RETURNS text AS
$BODY$
DECLARE
        return_text text = '';
        tmp_rec RECORD;
        parent_node text;
        where_clause text;
        object_prefix text;
        object_string text;
        group_text text = '';
        group_description text = '';
```

Retrieving details on function create_mib_notification_type... Done. 0.03 secs

EDB POSTGRES

What was wrong with it?

- C++ code dating back to 2002:
  - Code has grown messy over time
  - Very hard to find C++ developers
- Dated look and feel
- Desktop application in a web based world
- Dependent on troublesome third party libraries:

Open tickets reported by you

Can be seen only if you are logged in

Ticket	Summary	Type	Component	Patch	Status	Created	Modified
#4408	STC cannot handle long lines	defect	wxStyledText	False	confirmed	9 years	8 years
#4396	wxSTC AutoComplete menu broken with native wxListCtrl	defect	old wxOSX/Carbon port	False	confirmed	9 years	8 years
#4313	Numeric keypad doesn't work with wxSTC	defect	old wxOSX/Carbon port	False	confirmed	9 years	22 months
#4251	Keyboard navigation impossible	defect	wxAui	False	confirmed	9 years	8 years

How do we fix it?

- Change technology stack:
  - Easier to find developers
  - Could be web based
  - Much easier to change the look and feel
  - No complex and buggy cross-platform libraries

This means a complete rewrite!

pgAdmin 4

- Basic expectations
- Technology choice
- Non-functional requirements
- Functional requirements

Basic expectations

- The application should be able to run in both web and desktop modes
- The technology stack should be based on language(s) popular in the PostgreSQL community
- The core functionality of pgAdmin III should be re-implemented
- Features of pgAdmin III that are known to be used very little, if at all, should be excluded

Technology choice

- Python
  - Mature language
  - Used extensively in the postgresql.org infrastructure
  - Works well with PostgreSQL
- Javascript/jQuery/Bootstrap
  - Tried and tested technologies
  - Lots of developers with experience
- Flask micro-framework
  - A mature, yet lightweight Python web application framework
  - Very similar to the core of Django, which is well known in the community

Non-functional requirements (1)

- Framework:
  - The application should provide a framework for extensibility:
    - Treeview nodes are all plugins
    - Individual tools are plugins
    - Database drivers are plugins, to allow support for PostgreSQL derivatives
- Must be deployable:
  - In a desktop runtime, in single-user mode
  - On a web server using WSGI, in multi-user mode

Non-functional requirements (2)

- Packaging:
  - Windows/Mac packages for desktop deployment
  - Linux RPMs/DEBs, for desktop or server deployment
  - PIP wheel for server deployment
- Python compatibility:
  - All common versions of Python through to the latest
    - 2.7.x
    - 3.0.x – 3.5.x

Non-functional requirements (3)

- Key needs:
  - Use of one or more servers simultaneously
  - Support for all PostgreSQL datatypes
  - Support for UTF-8
  - i18n
- Speed:
  - Fast response, so it feels like a desktop application
  - No full page reloads; AJAX everywhere

Functional requirements (1)

- Support for all common database object types:
  - Servers, database, tablespaces, roles, extensions, schemas, tables, indexes, constraints, triggers, functions etc.
- Query Tool, with integrated data editing:
  - Merge the existing Query Tool and Edit Grid into a single tool
- Dashboards for simple realtime monitoring
- Procedural Language debugger
- Grant Wizard
- Backup and Restore

Functional requirements (2)

- Maintenance (VACUUM, ANALYZE etc)
- Utilities (Pause/resume WAL replay, add named restore point etc.)
- Online help for use of the application
- Links from object dialogues to relevant PostgreSQL documentation
- User manager with self-service password management for use in multi-user mode

pgAdmin4 – The Project

- The team
- Constraints
- Project management
- QA
- Release plan
- Stats

The team

- Almost all EDB staff, until we got the basics right (because I can boss them around ☺):
  - 3 developers/committers
  - 6 developers
  - 1 designer
  - 1 project manager
  - 4 quality assurance
  - 2 technical writers
  - 3 packagers
- Community
  - 15 contributors
  - ~75(?) bug reporters

Constraints

- Most of the work was being done by the Postgres Enterprise Manager development and QA teams:
  - Needs to start after PEM 6.0 is released
  - Needs to finish in time to start on PEM 7.0
  - Need to allow time for PEM update releases
  - Need to allow time for support escalations
- Code needed to be ready in time for inclusion in the EDB ‘one click’ PostgreSQL 9.6 installers

Project management

- Development schedule managed using traditional methods for ease of scheduling
  - projectmanager.com – multi-project/team aware online equivalent to Microsoft Project or OmniPlan
- Patch management for committers handled through a Kanban chart
  - Kanbanchi – online Kanban charts, integrated with Google Apps
- Post-development QA and bug tracking in the community
  - redmine.postgresql.org

projectmanager.com

secure.projectmanager.com

EnterpriseDB

Tutorials | Webinars | logout
Help | Support | Request Feature

Welcome | My Home | All | pgAdmin4 | New

MENU

Tasks | Dashboard | Files | Calendar | Issues | Expenses | Discuss | Email | Info

Save | Print | Share | Save to | Add Task | Cut | Copy | Paste | Undo | Redo | Delete | Assign | Indent | Outdent | Milestone | Attach | Notes | Comment | Link | Unlink | Color | Import | Export | Restore | Columns | Filter | Settings

All	Task Name	Planned Start Date	Planned Finish Date	Planned Duration	Planned Effort
71	Query Tool Layout	3/23/2016	3/31/2016	7 days	56 hours
72	EXPLAIN Mode and Options	4/1/2016	4/7/2016	5 days	30 hours
73	In-place Data Editing (where query allows)	3/9/2016	3/30/2016	16 days	96 hours
74	Query Execution	2/26/2016	3/7/2016	7 days	56 hours
75	Graphical EXPLAIN	1/29/2016	3/10/2016	30 days	180 hours
76	Integrate Graphical Explain in Query Tool	3/11/2016	3/15/2016	3 days	18 hours
77	Load/Save	3/18/2016	3/23/2016	4 days	32 hours
78	Recent Queries	3/11/2016	3/16/2016	4 days	32 hours
79	**Tools**	**1/29/2016**	**5/11/2016**	**74 days**	**736 hours**
80	Backup Globals	4/8/2016	4/29/2016	16 days	128 hours
81	Backup Server	3/16/2016	3/25/2016	8 days	64 hours
82	Backup Database	3/29/2016	4/7/2016	8 days	64 hours
83	Restore	5/2/2016	5/11/2016	8 days	64 hours
84	Import	4/8/2016	4/29/2016	16 days	128 hours
85	Maintenance Tool (VACUUM)	3/29/2016	4/6/2016	7 days	56 hours
86	Grant Wizard	2/4/2016	2/25/2016	16 days	96 hours
87	Options Dialogue	1/29/2016	2/22/2016	17 days	136 hours
88	**Utilities**	**3/31/2016**	**4/27/2016**	**20 days**	**132 hours**
89	Start/Stop/Reload/Restart Server	4/6/2016	4/8/2016	3 days	24 hours
90	Create Named Restore Point	4/7/2016	4/11/2016	3 days	18 hours
91	Pause/Resume WAL Replay	4/12/2016	4/14/2016	3 days	24 hours
92	Enable/Disable Triggers	4/15/2016	4/19/2016	3 days	18 hours

Mar, 28 '16 | Apr, 4 '16 | Apr, 11 '16 | Apr, 18 '16

Akshay Joshi (100%)
Surinder Kumar (100%)
Akshay Joshi (100%)
100%)
abuawala (100%)
Murtuza Zabuawala (100%)
Neel Patel (100%)
Neel Patel (100%)
Murtuza Zabuawala (100%)
Neel Patel (100%)
Murtuza Zab

Hide task info | Zoom:

General | Links | Resources | Notes | Files | Integrate | Dependencies

Share Plan: http://tinyurl.com/zbc4zd9

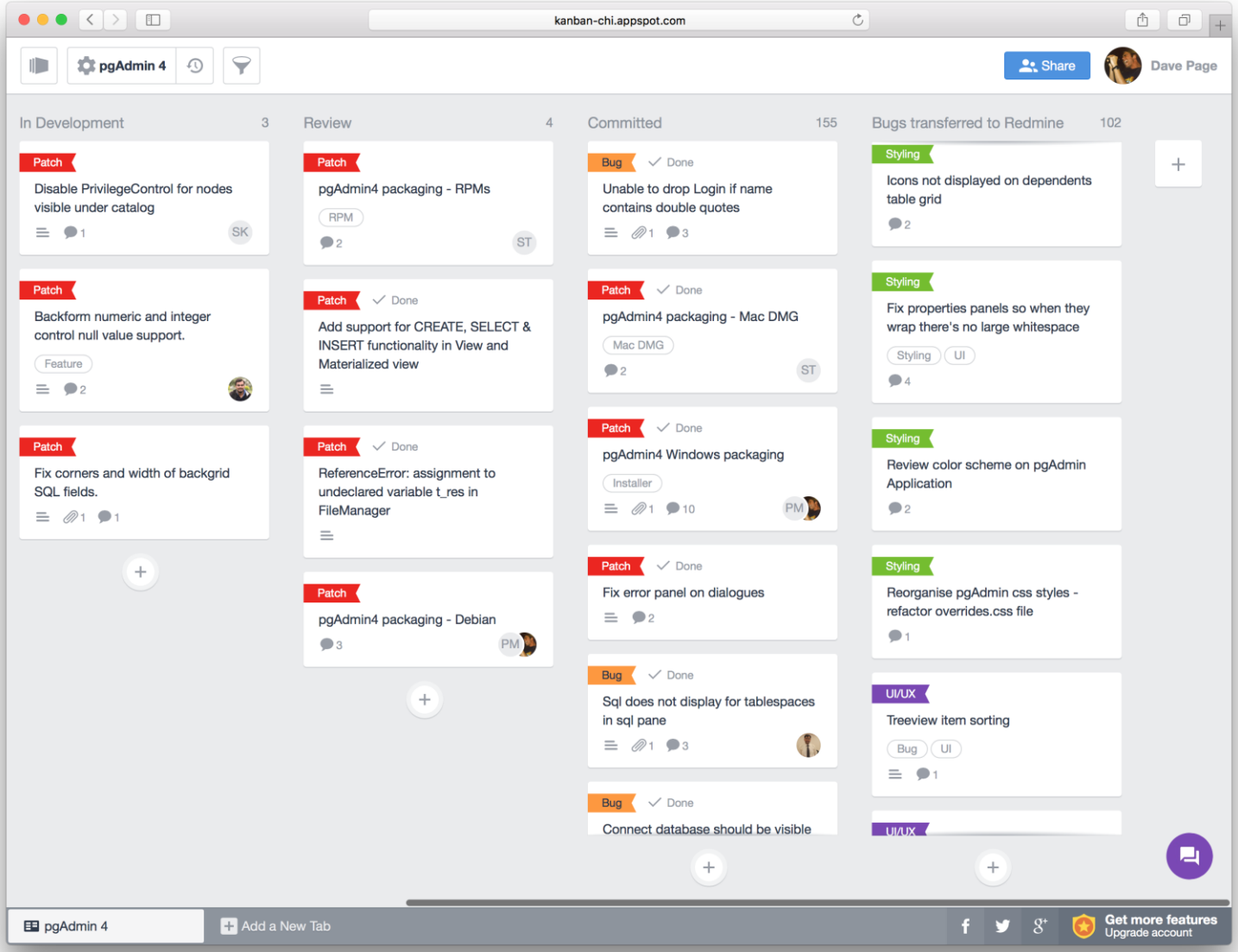
Name: Create Named Restore Point
Percent complete: 100% Priority: High
Locked
Milestone

Dates

Planned Start: 4/7/2016 Finish: 4/11/2016 Duration: 3 days Effort: 18 hours Cost:
Actual Start: 5/11/2016 Finish: 5/11/2016 Duration: 1 day Effort: Cost:
Baseline Start: 3/25/2016 Finish: 3/28/2016 Duration: 2 days Effort: 16 hours

EDB POSTGRES

Kanbanchi



Quality assurance

- Manual test suite developed by EDB QA Team:
  - Managed using Testrail QA
- Automated test framework by the QA team:
  - Built on the Python Unittest2 module
- Automated UI testing
  - Not yet underway
  - Will likely utilise Selenium
- Community testing
  - Ad-hoc; no guarantee of participants
  - But… have had mostly great feedback so far ☺
  - As well as some negative feedback ☹

Testrail QA

enterprisedb.testrail.net

← Return to Dashboard

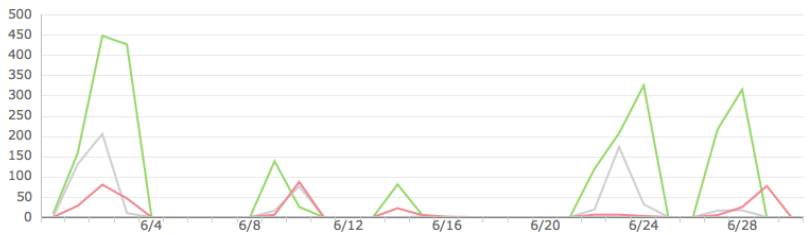
Working On | Dave Page | Help & Feedback

pgAdmin4

Overview | Todo | Milestones | Test Runs & Results | Test Suites & Cases | Reports

P26 pgAdmin4

Reports



In the past 30 days:

2470 Passed
69% set to Passed

696 Blocked
20% set to Blocked

0 Retest
0% set to Retest

397 Failed
11% set to Failed

Actions

Milestones
Add | View All

Test Runs
Add | View All

Test Suites
Add | View All

Todos

None.

Upcoming Milestones

Beta-3
No due date

Latest Test Runs

pgAdmin4 verification with PG servers
By Fahar A. on 6/24/2016

Detailed Tests - Sample
By Kanchan M. on 5/5/2016

Sanity Tests - Sample
By Kanchan M. on 5/5/2016

Activity — History | Test Changes

Wednesday, June 29, 2016

Milestone	Beta-2 (pgAdmin4 website Installers) (completed)	Completed by Fahar A.
Test Plan	pgadmin4-1.0-beta2 testing (completed)	Closed by Fahar A.
Milestone	~~pgAdmin4 verification with PG server~~	Deleted by Kanchan M.
Milestone	Beta-3	Created by Kanchan M.

Monday, June 27, 2016

Milestone	preBeta (completed)	Completed by Priyanka S.
Test Plan	pgadmin4-1.0-beta2 testing (completed)	Created by Fahar A.
Milestone	Beta-2 (pgAdmin4 website Installers) (completed)	Created by Fahar A.
Milestone	Beta-2 (pgAdmin4 with Postgres 9.6 Installers) (completed)	Completed by Fahar A.

EDB POSTGRES

Regression tests

~/git/pgadmin4/web/regression — dpage@borg:~ — -bash

```
Ran 145 tests in 31.493s

OK

======================================================================
Test Result Summary
======================================================================

Regression - EPAS 9.5:

        145 tests passed
        0 tests failed
        0 tests skipped

Regression - PG 9.5:

        133 tests passed
        0 tests failed
        12 tests skipped:
                PackageAddTestCase
                PackageDeleteTestCase
                PackageGetTestCase
                PackagePutTestCase
                SynonymAddTestCase
                SynonymDeleteTestCase
                SynonymGetTestCase
                SynonymPutTestCase
                ResourceGroupsAddTestCase
                ResourceGroupsDeleteTestCase
                ResourceGroupsPutTestCase
                ResourceGroupsGetTestCase

Regression - PG 9.4:

        133 tests passed
        0 tests failed
        12 tests skipped:
                PackageAddTestCase
                PackageDeleteTestCase
                PackageGetTestCase
                PackagePutTestCase
                SynonymAddTestCase
                SynonymDeleteTestCase
                SynonymGetTestCase
                SynonymPutTestCase
                ResourceGroupsAddTestCase
                ResourceGroupsDeleteTestCase
                ResourceGroupsPutTestCase
                ResourceGroupsGetTestCase

======================================================================

Please check output in file: /Users/dpage/git/pgadmin4/web/regression/regression.log

(pgadmin4)piranha:regression dpage$
```

Release timeline

- Beta 1 released on 7th June
- Beta 2 released on 24th June (PostgreSQL 9.6 Beta 2)
- Beta 3 released on 21st July (PostgreSQL 9.6 Beta 3)
- Beta 4 released on 18th August (PostgreSQL 9.6 Beta 4)
- RC 1 released on 1st September (PostgreSQL 9.6 RC 1)
- v1.0 released on 29th September (PostgreSQL 9.6.0)
- v1.1 released on 27th October (PostgreSQL 9.6.1):
  - 39 bug fixes & 2 new features
- V1.2 released on… TBD!
  - 25 bug fixes & 3 new features, as of 2016-11-22

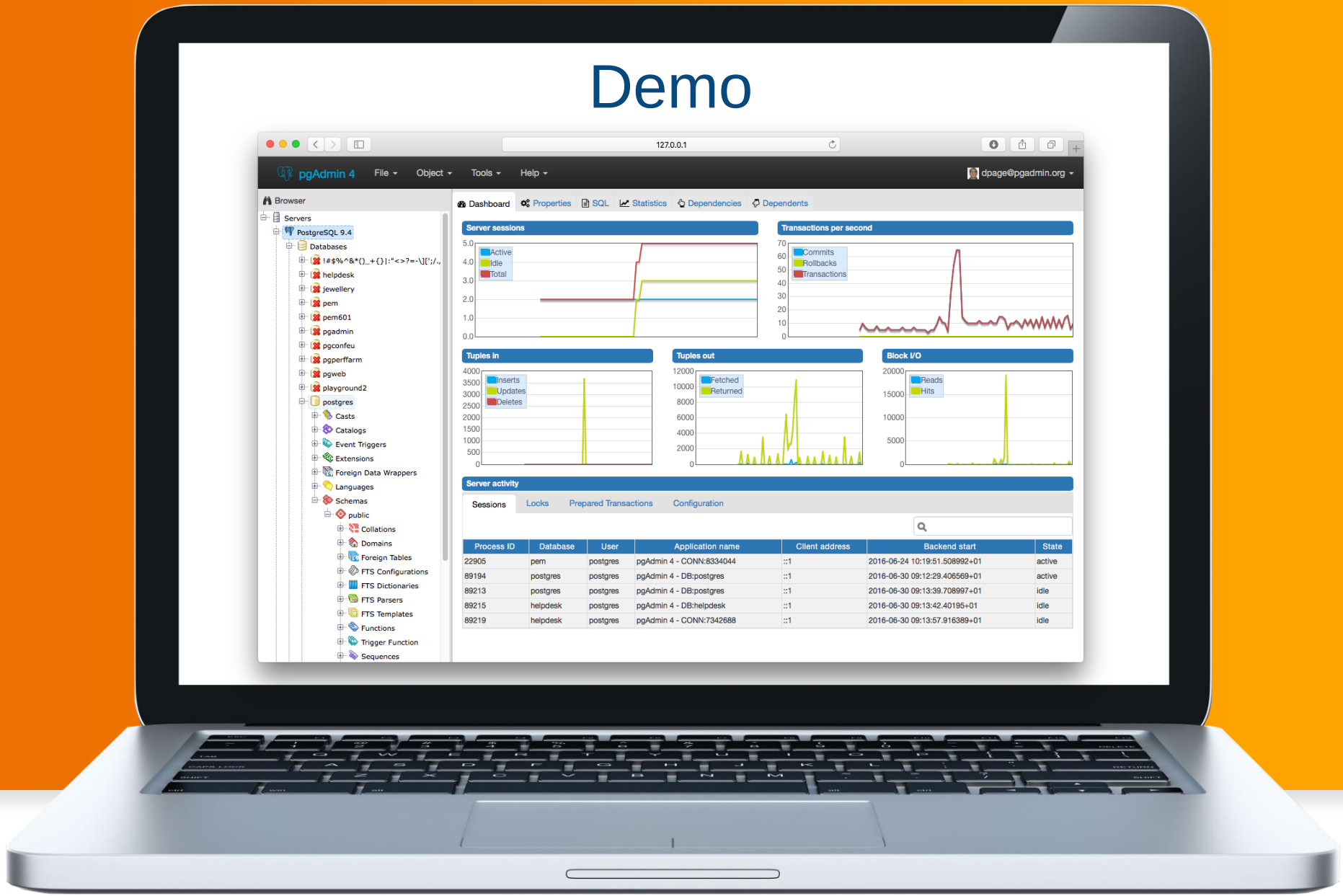
Stats (as of 22nd November)

- 20 EDB contributors, totalling ~15,000 hours of effort
- ~90 community contributors
- 1555 commits
- 108 bugs outstanding
- 20 bug fixes in QA
- 69 feature requests
- 380 bugs confirmed resolved by QA

Stats (as of 22nd November)

- 239,008 lines of code:
  - 149,715 – Javascript (inc. libraries)
  - 42,979 – Python
  - 21,174 – CSS (inc. libraries)
  - 16,107 – SQL
- 6,307 lines of documentation source:
  - 87 RST files
  - 304 screen shots

Demo



EDB POSTGRES

More info

- Website:

  https://www.pgadmin.org/

- Source code:

  https://git.postgresql.org/gitweb/?p=pgadmin4.git

- Mailing lists:

  pgadmin-support@postgresql.org

  pgadmin-hackers@postgresql.org

EDB POSTGRES